Introduction

In the first CS112 homework assignment, we spent a lot of time and effort setting up GitHub repositories, installing git software, and synchronizing directories on our personal computers with GitHub repositories. With the setup complete, we can complete the rest of the CS112 class using only a small set of git commands.
This document summarizes the git commands we use every day, and it introduces a few other commands that might be useful. Finally, it proposes some optional configuration we can do on Mac computers to keep our repositories cleaner.

Basic Git Commands

Here are the git commands we use every day to update our GitHub repositories and also to pull information from the CourseInfo repository to our local computers.

command	usage	comments
git add	git add <<list of files to add>>	This command tells git to add our list of files (which can be only one file, multiple files, or a directory name) to the "latest version."
git commit	git commit -m "comments for this version"	This command tells git to "freeze" the current version, with whatever files have been added via "git add".
git push	git push	This command pushes any completed versions to the GitHub repository
git pull	git pull	This command finds any changes that have been made to the GitHub cloud repository that have not yet been pulled to our computer, and it pulls them to our computer. This should e used mainly with the CourseInfo repository, where the instructors push new information to GitHub.

A Few More Git Commands

Here are a few more git commands you might find useful.

Command	Usage	Comments
git log	git log	Prints a list of all versions that have been made in the current repository, along with whatever comments were entered with the “git commit” command
git status	git status	Prints any changes in the local files that have not been “git added” yet, and any changes that have been added that have not been “git committed” yet.
git checkout	git checkout	Suppose you pulled some files into your PC but you moved or deleted them and want them back. “git pull” will not get them, since it only looks for changes to the GitHub repository, not to your local computer. “git checkout” retrieves files from the GitHub repository even if they have been pulled before

"git push" doesn't work!

Only change your repository from one place. I recommend the terminal window.

See your professor for help with a broken setup.

Mac-specific Configuration

The MacOS operating system on Apple computers repeatedly creates subdirectories called “.DS_Store” in every directory in your filesystem. It is possible for these directories to mess up git, if they get pushed to GitHub and change and get out-of-sync between GitHub and your computer. Wouldn’t it be nice if git would simply ignore the .DS_store directories?

That is possible. In your home directory (e.g. /Users/stephcurry) you can put a file called “.gitignore” that contains the single line:

```
.DS_store
```

Git will read this file and then know not to pay any attention to files and directories called “.DS_store”. To enforce this on git immediately, you can run the following command:

```
git config --global core.excludesfile ~/.gitignore
```